AMENDMENTS TO THE CLAIMS

1. (currently amended): A method comprising:

emulating an operation of a client; and

permitting the emulated operation to access a contiguous portion of emulated memory only when a pointer used by the emulated operation and a table entry used to manage the emulated memory both contain the same identifier, wherein an address to the contiguous portion is contained in both the pointer and the table entry and wherein the identifier is removed from the corresponding pointer to permit the access to the contiguous portion of emulated memory.

2. (original): The method as defined in Claim 1, wherein:

the table entry is in a table that contains a plurality of said table entries;

each said table entry references an address of one said contiguous portion of the emulated memory;

the pointer is one of a plurality of said pointers; and

each said pointer contains:

the address of a respective said contiguous portion of the emulated memory; and

one said identifier corresponding to the respective said contiguous portion of the emulated memory.

3. (currently amended): The method as defined in Claim 1, wherein the permitted access further comprises [[:]]

~~removing the identifier from the corresponding pointer to the contiguous portion of emulated memory; and~~

when the permitted access is not a read or a write operation, identically changing the identifier in both of the corresponding pointer to contiguous portion of emulated memory and the corresponding table entry.

4. (original): The method as defined in Claim 1, wherein the client is selected from the group consisting of:

a personal computer (PC);

a workstation;

a server;

a set top box;

a video game console;

a Personal Digital Assistant (PDA);

a cellular telephone;

a handheld computing device; and

a computing device having less memory and/or computing resources than that of another computing device executing an application that emulates the operation of the client.

5. (original): A computer-readable medium comprising instructions that, when executed by a computer, performs the method of Claim 1.

6. (currently amended): A method comprising:

making a call to a memory manager for an emulated memory access operation to an allocated contiguous portion of emulated memory, wherein a generation count has been assigned to:

a plurality of table entries corresponding to a respective plurality of said allocated contiguous portions of emulated memory, and

a plurality of pointers each containing an address to a respective said allocated contiguous portion of emulated memory;

comparing the generation count:

in the pointer containing the address to the allocated contiguous portion of emulated memory; and

in the table entry corresponding to the allocated contiguous portion of emulated memory;

if the respective said generation counts in the comparison do not match, then outputting a diagnostic; and

if the respective said generation counts in the comparison match, removing the generation count from the pointer specified by the memory manager for the

emulated memory access operation during the performing of the emulated memory access operation for which the memory manager was called.

7. (original): The method as defined in Claim 6, further comprising:

performing the emulated memory access operation for which the memory manager was called when there is a match of the respective said generation counts; and

preventing the performance of the emulated memory access operation for which the memory manager was called when the respective said generation counts of the comparison do not match.

8. (original): The method as defined in Claim 7, further comprising, when there is a match and the emulated memory access operation is not a read or a write operation, incrementing the generation count in both:

the pointer containing the address to the allocated contiguous portion of emulated memory; and

the table entry corresponding to the allocated contiguous portion of emulated memory.

9. (cancelled).

10. (original): The method as defined in Claim 6, wherein the emulated memory access operation is selected from the group consisting of:

a read operation;

a write operation;

a reallocation operation; and

an operation to free one or more of said allocated contiguous portions of emulated memory.

11. (original): The method as defined in Claim 6, further comprising, prior to the making of the call:

making a call to the memory manager for to allocate a contiguous portion of emulated memory;

receiving one said pointer from the memory manager that contains the address of the allocated contiguous portion of emulated memory;

performing the allocation of the contiguous portion of emulated memory; and

inserting the generation count:

in the:

the pointer containing the address to the one said allocated contiguous portion of emulated memory; and

the plurality of table entries corresponding to the one said allocated contiguous portion of emulated memory.

12. (original): A first apparatus for emulating a second apparatus, wherein:

the first apparatus performs the method of Claim 6, and

the second apparatus is selected from the group consisting of:

a personal computer (PC);

a workstation;

a server;

a set top box;

a video game console;

a PDA;

a cellular telephone;

a handheld computing device; and

a client having less memory and/or computing resources than that of the first apparatus.

13. (original): A computer-readable medium comprising instructions that, when executed by a computer, performs the method of Claim 12.

14. (currently amended): In a first computing device executing a first application for the emulation of a second computing device executing a second application, a method comprising:

making a call from the second application to a memory manager for an emulated memory access operation to an allocated contiguous portion of emulated memory used by the second application and including a plurality of said allocated contiguous portions, wherein:

a generation count is in a plurality of table entries corresponding to a respective plurality of said allocated contiguous portions of emulated memory;

a generation count is in a plurality of pointers each containing an address to a respective said allocated contiguous portion of emulated memory;

for the emulated memory access operation, the memory manager uses the address in the pointer that corresponds to the allocated contiguous portion in emulated memory after removal of the generation count from the pointer; and

prior to performing the emulated memory access operation to the allocated contiguous portion of emulated memory:

comparing the generation count:

in the pointer containing the address of the allocated contiguous portion of the emulated memory; and

in the table entry corresponding to the allocated contiguous portion of the emulated memory;

outputting a diagnostic when the respective said generation counts of the comparison do not match.

15. (original): The method as defined in Claim 14, further comprising:

performing the emulated memory access operation for which the memory manager was called when there is a match of the respective said generation counts; and

preventing the performance of the emulated memory access operation for which the memory manager was called when the respective said generation counts of the comparison do not match.

16. (original): The method as defined in Claim 15 further comprising, when there is a match of the respective said generation counts and the emulated memory access operation is not a read operation or a write operation, incrementing the generation count in both:

the pointer containing the address to the allocated contiguous portion of emulated memory; and

the table entry corresponding to the allocated contiguous portion of emulated memory.

17. (original): The method as defined in Claim 14, further comprising, when:

the comparison finds that there is a match of the respective said generation counts; and

the emulated memory access operation is neither a read operation nor a write operation:

performing the emulated memory access operation for which the memory manager was called and during which the generation count is removed from the pointer used by the memory manager.

18. (original): The method as defined in Claim 14, wherein the emulated memory access operation is selected from the group consisting of:

a read operation;

a write operation;

a reallocation operation; and

an operation to free one or more of said allocated contiguous portions of emulated memory.

19. (original): The method as defined in Claim 14, further comprising, prior to the making of the call by the second application to the memory manager for the emulated memory access operation:

making a call by the second application to the memory manager for an allocation of said allocated contiguous portion of emulated memory;

receiving one said pointer from the memory manager that contains an address to said allocated contiguous portion of emulated memory;

performing an allocation of said allocated contiguous portions of emulated memory; and

incrementing the generation count in both:

the pointer containing the address to said allocated contiguous portion of emulated memory; and

the table entry corresponding to said allocated contiguous portion of emulated memory.

20. (original): The method as defined in Claim 14, wherein the second computing device is selected from the group consisting of:

a PC;

a workstation;

a server;

a set top box;

a video game console;

a PDA;

a cellular telephone;

a handheld computing device;

a consumer electronic device having a processor and memory; and

a client having less memory and/or computing resources than that of the first computing device.

21. (original): A computer-readable medium comprising instructions that, when executed by a computer, performs the method of Claim 14.

22. (original): A computer-readable medium containing instructions for execution by a computer, wherein the instructions comprise:

first logic calling for an emulated memory access operation with respect to a first of a contiguous portion of an emulated memory for which there is:

a corresponding table entry in a table having a plurality of said table entries that map to respective other said portions of the emulated memory, wherein each said table entry contains an identifier; and

a corresponding pointer to a plurality of pointers each containing an identifier and an address to a respective said contiguous portion of the emulated memory;

second logic, in response to the first logic, such that, if the identifier in the table entry corresponding to the first said contiguous portion is the same as the identifier in the pointer corresponding to the first said portion, then:

the emulated memory access operation is performed with respect to the first said contiguous portion of the emulated memory; and

when the emulated memory access operation is neither a read operation nor a write operation, the identifier is identically changed in both:

the table entry corresponding to the first said portion; and

the pointer corresponding to the first said portion;

third logic, when the identifier in the table entry corresponding to the first said contiguous portion is different from the identifier in the pointer corresponding to the first said portion, calling for a diagnostic to be output.

23. (original): The computer-readable medium as defined in Claim 22, wherein the emulated memory access operation is selected from the group consisting of:

a read operation;

a write operation;

a reallocation operation; and

an operation to free one or more of said portions of the emulated memory.

24. (original): The computer-readable medium as defined in Claim 22, wherein the performance of the memory operation further comprises removing the identifier from the pointer corresponding to the first said contiguous portion during the performance of the memory operation.

25. (original): A first apparatus to execute each said logic of Claim 22 so as to emulate a second apparatus executing an application using the emulated memory, wherein the second apparatus is selected from the group consisting of:

a PC;

a workstation;

a server;

a set top box;

a video game console;

a PDA;

a cellular telephone;

a handheld computing device; and

a client having less memory and/or computing resources than that of the first apparatus.

26. (currently amended): A first software program which, when executed by a computing device, emulates the execution of a second software program using emulated memory, the first software program comprising instructions that permit the second software program to perform an emulated memory access operation on a previously allocated contiguous portion of the emulated memory only when a pointer and a table entry both contain the same identifier, wherein:

the pointer also contains an address to the previously allocated contiguous portion which is useable to access the previously allocated contiguous portion after removal of the identifier; and

the table entry maps to the previously allocated contiguous portion.

27. (original): The first software program as defined in Claim 26, wherein.

the table entry is one of a plurality of said table entries that map to a respective plurality of said portions of the emulated memory

the pointer is one of a plurality of said pointers that each contain:

the address to a respective said contiguous portion of the emulated memory; and

one said identifier corresponding to the respective said contiguous portion of the emulated memory.

28. (original): The first software program as defined in Claim 26, wherein

the performance of the emulated memory access operation on the contiguous portion of the emulated memory further comprises:

removing the identifier from the corresponding pointer when it is processed by the execution of the second software program; and

when the emulated memory access operation is neither a read operation not a write operation, identically changing the identifier with the first software program in both of the corresponding pointer and table entry after the execution of

the second software program has performed the emulated memory access operation on the contiguous portion of the emulated memory.

29. (original): The first software program as defined in Claim 27, wherein the instructions further comprise removing the identifier from each said pointer prior to its use by the second software program.

30. (original): The first software program as defined in Claim 27, wherein the instructions further comprise use of the table entries and identifiers with the first software program but not by the second software program.

31. (original): A first apparatus to execute the first software program as defined in Claim 26, and thereby emulate a second apparatus executing the second software program, wherein the second apparatus is selected from the group consisting of:

a PC;

a workstation;

a server;

a set top box;

a video game console;

a PDA;

a cellular telephone;

a handheld computing device; and

a client having less memory and/or computing resources than that of the first apparatus.

32. (currently amended): A computer-readable medium containing instructions for execution by a computer, wherein the instructions comprise:

means for emulating an operation of a client as the client executes an application; and

means for outputting a diagnostic when:

the emulated operation attempts to access a previously allocated contiguous portion of emulated memory using a pointer containing an identifier, wherein the pointer is configured to access the previously allocated contiguous portion of the emulated memory upon removal of the identifier; and

a table entry used to manage the emulated memory does not contain the same identifier as the identifier in the pointer, wherein an address to the previously allocated contiguous portion is contained in both the pointer and the table entry.

33. (original): The computer-readable medium as defined in Claim 32, wherein:

the table entry is in a table that contains a plurality of said table entries;

each said table entry references an address of one said previously allocated contiguous portion of the emulated memory;

the pointer is one of a plurality of said pointers; and

each said pointer contains:

the address to a respective said previously allocated contiguous portion of the emulated memory; and

one said identifier corresponding to the respective said previously allocated contiguous portion of the emulated memory.

34. (original): The computer-readable medium as defined in Claim 32, further comprising means for permitted the attempted access by the emulated operation to the previously allocated contiguous portion of emulated memory, wherein during prior to said access:

the identifier is removed from the corresponding pointer to the contiguous portion of emulated memory; and

when the permitted access is not a read or a write operation, the identifier in both of the corresponding pointer to contiguous portion of emulated memory and the corresponding table entry is identically changed.

35. (original): The computer-readable medium as defined in Claim 34, further comprising:

means, prior to an allocation of the previously allocated contiguous portion of emulated memory, for making a call to a memory manager for an allocation of the previously allocated contiguous portion of emulated memory;

means for receiving the pointer from the memory manager that contains the address to the previously allocated contiguous portion of emulated memory;

means for performing the allocation of the previously allocated contiguous portion of emulated memory;

means for inserting the generation count in the table entry; and

means for copying the generation count from the table entry to the pointer.

36. (original): The computer-readable medium as defined in Claim 32, wherein the client being emulated is selected from the group consisting of:

a PC;

a workstation;

a server;

a set top box;

a video game console;

a PDA;

a cellular telephone;

a handheld computing device; and

a computing device having less memory and/or computing resources than that of another computing device executing an application that emulates the operation of the client.